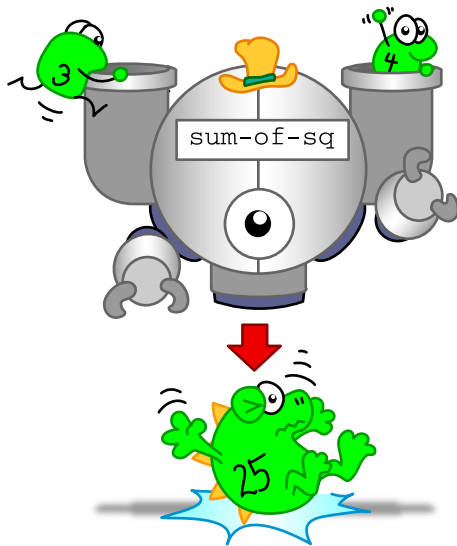# Functions as Data
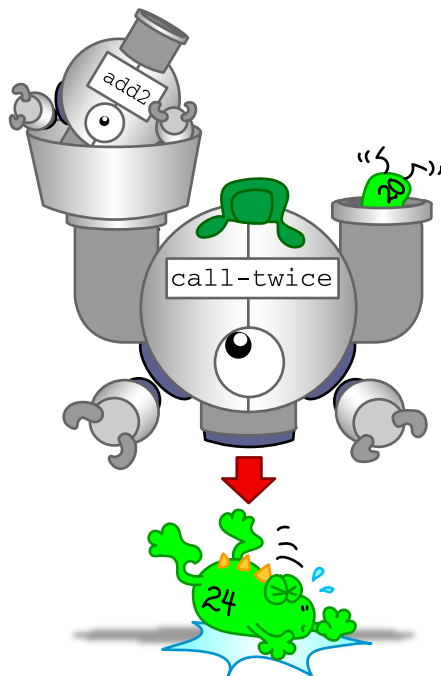## The secret to Scheme's success

Most functions take in data as arguments and output data.


```
;; Compute sum of squares
> (define (sum-of-sq x y)
      (+ (* x x) (* y y)))
sum-of-sq

> (sum-of-sq 3 4)
25
```

But Scheme is great because it's easy to pass <u>functions</u> as arguments, like they're data!
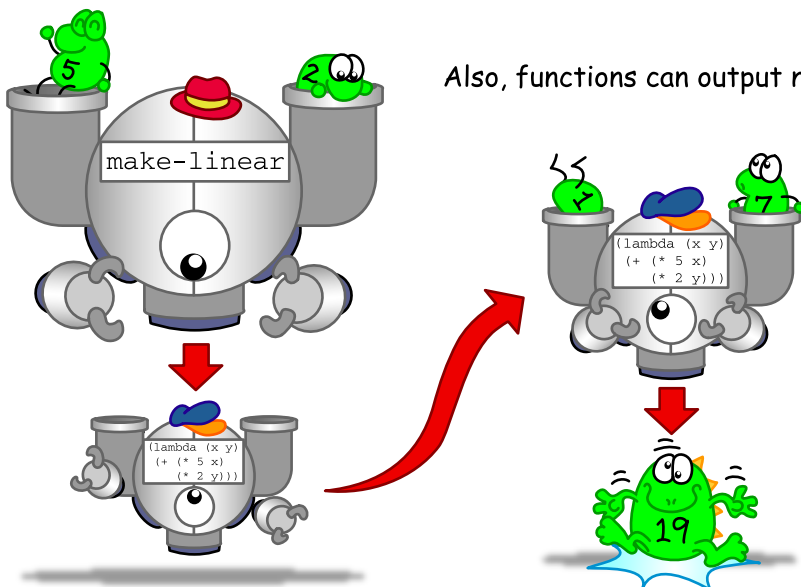


```
> (define (add2 n) (+ n 2))
add2

;; Invoke a function twice
> (define (call-twice func x)
      (func (func x)))
call-twice

> (call-twice add2 20)
24
```

Also, functions can output new functions!


```
;; Generate a linear equation function
> (define (make-linear a b)
      (lambda (x y) (+ (*a x) (* b y))))
make-linear

> ((make-linear 5 2) 1 7)
19
```

Katrina Yim